

WHAT IS CLAIMED IS:

1. A system for validating an interface of a dynamically  
2 linkable component, comprising:

3 a check code generator that transforms said interface of said  
4 dynamically linkable component into an interface identifier  
5 representing said interface and couples said interface identifier  
6 to said dynamically linkable component; and

7 a interface verifier that employs said interface identifier to  
8 determine a compatibility of said dynamically linkable component.

2. The system as recited in Claim 1 wherein said check code  
generator transforms said interface of said dynamically linkable  
component into said interface identifier by transforming a textual  
representation of at least a portion of said interface.

3. The system as recited in Claim 1 wherein said check code  
generator couples said interface identifier to said dynamically  
linkable component by placing said interface identifier in a types  
declaration file.

4. The system as recited in Claim 1 wherein said interface  
2 identifier varies as a function of a version of said dynamically  
3 linkable component.

5. The system as recited in Claim 1 wherein said interface  
2 verifier employs said interface identifier to determine a  
3 compatibility of said dynamically linkable component with a second  
4 dynamically linkable component.

6. The system as recited in Claim 1 wherein said interface  
verifier is a part of a second dynamically linkable component.

7. The system as recited in Claim 1 wherein said interface  
verifier determines a compatibility of said dynamically linkable  
component by comparing said interface identifier with a history  
list containing at least one member.

8. The system as recited in Claim 1 wherein said interface  
2 identifier is a type selected from the group consisting of:  
3 a check sum, and  
4 a cyclic redundancy check.

9. The system as recited in Claim 1 wherein said check code  
2 generator uses filtering directives to include and exclude portions  
3 of said interface from said interface identifier.

TELETYPE - SOURCE CODE

10. A method of validating an interface of a dynamically

2 linkable component, comprising:

3 transforming said interface of said dynamically linkable  
4 component into an interface identifier representing said interface;

5 coupling said interface identifier to said dynamically  
6 linkable component; and

7 employing said interface identifier to determine a  
8 compatibility of said dynamically linkable component.

11. The method as recited in Claim 10 wherein said  
transforming comprises transforming a textual representation of at  
least a portion of said interface.

12. The method as recited in Claim 10 wherein said coupling  
comprises placing said interface identifier in a types declaration  
file.

13. The method as recited in Claim 10 wherein said interface

2 identifier varies as a function of a version of said dynamically  
3 linkable component.

14. The method as recited in Claim 10 wherein said employing  
2 comprises employing said interface identifier to determine a  
3 compatibility of said dynamically linkable component with a second  
4 dynamically linkable component.

15. The method as recited in Claim 10 wherein said interface  
2 verifier is a part of a second dynamically linkable component.

16. The method as recited in Claim 10 wherein said employing  
2 comprises comparing said interface identifier with a history list  
3 containing at least one member.

17. The method as recited in Claim 10 wherein said interface  
2 identifier is a type selected from the group consisting of:  
3 a check sum, and  
4 a cyclic redundancy check.

18. The method as recited in Claim 10 wherein said  
2 transforming uses filtering directives to include and exclude  
3 portions of said interface from said interface identifier.

*Stop  
Off*

19. A system for validating an interface of a dynamically

2 linkable component, comprising:

3 an interface identifier, coupled to said dynamically linkable

4 component, that represents said interface of said dynamically

5 linkable component; and

6 a interface verifier that employs said interface identifier to

7 determine a compatibility of said dynamically linkable component.

20. The system as recited in Claim 19 wherein said interface

identifier is contained within a types declaration file.

21. The system as recited in Claim 19 wherein said interface

identifier varies as a function of a version of said dynamically

linkable component.

22. The system as recited in Claim 19 wherein said interface

2 verifier employs said interface identifier to determine a

3 compatibility of said dynamically linkable component with a second

4 dynamically linkable component.

23. The system as recited in Claim 19 wherein said interface

2 verifier is a part of a second dynamically linkable component.

24. The system as recited in Claim 19 wherein said interface  
2 verifier determines a compatibility of said dynamically linkable  
3 component by comparing said interface identifier with a history  
4 list containing at least one member.

*25*  
*26*  
25. A method of validating an interface of a dynamically linkable component, comprising:

3. coupling an interface identifier to said dynamically linkable  
4. component; and  
5. employing said interface identifier to determine a  
6. compatibility of said dynamically linkable component.

26. The method as recited in Claim 25 wherein said coupling  
comprises placing said interface identifier in a types declaration  
file.

27. The method as recited in Claim 25 wherein said interface identifier varies as a function of a version of said dynamically linkable component.

28. The method as recited in Claim 25 wherein said employing  
comprises employing said interface identifier to determine a  
3. compatibility of said dynamically linkable component with a second  
4. dynamically linkable component.

29. The method as recited in Claim 25 wherein said interface verifier is a part of a second dynamically linkable component.

30. The method as recited in Claim 25 wherein said employing  
2 comprises comparing said interface identifier with a history list  
3 containing at least one member.

*Sub  
A/C*

2 31. A real-time process control system, comprising:  
a plurality of sensors and controllable devices;  
3 a controller, coupled to said plurality of sensors and  
4 controllable devices, that executes software having at least first  
5 and second dynamically linkable components to coordinate an  
6 operation of said plurality of sensors and controllable devices;  
7 an interface identifier, coupled to said first dynamically  
8 linkable component, that represents an interface of said first  
9 dynamically linkable component; and  
10 a interface verifier that employs said interface identifier to  
11 determine a compatibility of said first and second dynamically  
12 linkable components.

32. The real-time process control system as recited in Claim  
31 wherein said interface identifier is a transformation of a  
textual representation of at least a portion of said interface.

2 33. The real-time process control system as recited in Claim  
31 wherein said interface identifier is contained within in a types  
3 declaration file.

34. The real-time process control system as recited in Claim  
2 31 wherein said interface identifier varies as a function of a  
3 version of said first dynamically linkable component.

35. The real-time process control system as recited in Claim  
2 31 wherein said interface verifier is a part of said second  
3 dynamically linkable component.

36. The real-time process control system as recited in Claim  
2 31 wherein said interface verifier determines a compatibility of  
said first dynamically linkable component by comparing said  
interface identifier with a history list associated with said  
second dynamically linkable component and containing at least one  
member.

37. The real-time process control system as recited in Claim  
2 31 wherein said interface identifier is a type selected from the  
3 group consisting of:  
4 a check sum, and  
5 a cyclic redundancy check.